Preprint
UCRL-JC-143464

Data Mining for Security Information: A Survey

S. T. Brugger, M. Kelley, K. Sumikawa, S. Wakumoto

This article was submitted to
8th Association for Computing Machinery Conference on Computer & Communications Security, Philadelphia, PA., November 6-8, 2001

April 19, 2001

U.S. Department of Energy
Lawrence Livermore National Laboratory

This work was performed under the auspices of the United States Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

Data Mining for Security Information: A Survey

S Terry Brugger <zow@llnl.gov>

Marcey Kelley <kelley6@llnl.gov>

Ken Sumikawa <sumikawa2@llnl.gov>

Shaun Wakumoto <wakumoto1@llnl.gov>

Mar 5, 2001

Last revised: Apr 19, 2001

This paper will present a survey of the current published work and products available to do off-line data mining for computer network security information.

Introduction

Hundreds of megabytes of data are collected every second that are of interest to computer security professionals. This data can answer questions ranging from the proactive, "Which machines are the attackers going to try to compromise?" to the reactive, "When did the intruder break into my system and how?" Unfortunately, there's so much data that computer security professionals don't have time to sort through it all. What we need are systems that perform data mining at various levels on this corpus of data in order to ease the burden of the human analyst. Such systems typically operate on log data produced by hosts, firewalls and intrusion detection systems as such data is typically in a standard, machine readable format and usually provides information that is most relevant to the security of the system.

Systems that do this type of data mining for security information fall under the classification of intrusion detection systems. It is important to point out that we are not surveying real-time intrusion detection systems. Instead, we examined what is possible when the analysis is done off-line. Doing the analysis off-line allows for a larger amount of data correlation between distant sites who transfer relevant log files periodically and may be able to take greater advantage of an archive of past logs. Such a system is not a replacement for a real-time intrusion detection system but should be used in conjunction with one. In fact, as noted previously, the logs of the real-time IDS may be one of the inputs to the data mining system.

We will concentrate on the application of data mining to network connection data, as opposed to system logs or the output of real-time intrusion detection systems. We do this primarily because this data is readily obtained from firewalls or real-time intrusion detectors and it looks the same regardless of the network architecture or the systems that run on the network. This similarity greatly simplifies the data cleansing step and provides a dataset with high orthogonality between multiple sites, increasing the accuracy of the data mining operations. The decision to use connection logs instead of packet logs is discussed below.

This paper will survey both the research that has been done in this area to date and publicly available products that perform such tasks.

Research

A great deal of research has been done on the application of data mining to security information. Much of the work wasn't as concerned with using general data mining techniques as applying specific techniques that are typically associated with knowledge discovery to logged information. These techniques ranged from genetic algorithms to fuzzy profiling. We will examine what the goal of data mining is, what an ideal system for applying data mining techniques to security information would do, the architecture for such a system, and present what research has been done on particular techniques as well as the results of that research.

Goal of data mining

In order to figure out how data mining can be applied to find relevant computer security information, we must first define what data mining is. Generally, data mining is the process of extracting useful models or patterns from large data stores that may have previously gone unnoticed [Bass, Lee:framework].

The data mining operations may result in statistical analysis, deviation analysis, rule abduction, neural abduction, making associations, correlations, link (or tree) abduction, sequence abduction, and clustering (also known as classification) [Waltz , T&C, Lee:framework Lee:thesis]. The last three have been identified as being the most useful to network security [Lee: framework] and clustering or classification is the most frequently cited goal [Ghosh, Kumar].

Given all these techniques, we hope to to find "hidden patterns based on previously undetected intrusions to help develop new detection templates," [Bass]. This allows us to transcend the limitations of many current IDS which rely on a static set of intrusion signatures (misuse detection systems) and "evolve from memorization to generalization," [Ghosh]. Such a system would be a type of anomaly detection system. "Anomaly detection attempts to quantify the usual or acceptable behavior and flags other irregular behavior as potentially intrusive," [Kumar]. The first example of such a system was IDES as described in Denning's seminal 1987 paper [Denning], which focused on mining statistical measures to use for comparison when searching for anomalies. By using more advanced data mining techniques it is hoped that additional attacks may be detected and the false positive (false alarm) rate will decrease.

An ideal system

The real power of applying data mining to log information is the potential to move beyond intrusion detection. Intrusion detection today is mostly a matter of "Close the barn door Edith, the cows just escaped!" [craw]. An advanced system should be able to perform attack prediction and threat analysis [Bass, T&C]. Bass notes that the output of an ideal system "would be estimates of the identity (and possibly the location) of an intruder, the intruder's activity, the observed threats, the attack rates, and an assessment of the severity of the cyberattack," [Bass]. Lee notes in that "a typical attack session can be split into three phases: a learning phase, a standard attack phase, and an innovative attack phase," [Lee:thesis]. We should be able to use that fact to classify malicious traffic, determine the level of risk and forecast future attacks.

Such a system may advance well beyond current ID systems. Eventually such advanced systems may advance to the point that we can "identify and track multiple hostile information flows for targets, attack rate, and severity in cyberspace," [Bass]. It will accomplish such a task though the use of self-training, applying "knowledge acquired in one learning task to another learning task in order to speed learning and possibly increase final accuracy," [Lane]. In the end, it will likely bear a good resemblance to a biological system. Some research has already likened an IDS to the human immune system [UNM].

System Architecture

The general architecture of a system that performs data mining for security information will most likely look like a hybrid between a generic data mining system and a distributed IDS. Bass presents both high level architectures for data fusion and mining in [Bass], although we envision that a system that does data mining for security knowledge will incorporate all the data fusion features. Dickerson & Dickerson present a simple three tiered architecture to do data mining (with a focus on fuzzy analysis, described below) in [Dickerson]. The first tier is made up of numerous Network Data Collectors which just grab packets off the wire and feed them into the second tier. This second tier is made up of numerous Network Data Processors, which process the packets for the information that's needed at the third tier. This third tier is a centralized Fuzzy Threat Analyzer, which applies the technique described below to ascertain the risk presented by a given connection.

The actual data collection can take place in whatever manner is most apropos for the given environment. The data should be cleansed as it enters the database (or as it gets transferred from a "raw data" table to a "cleansed data" table). The most popular standard format to do analysis on is the connection log. Besides being readily available and a much more reasonable size than other log formats (like packet logs), the connection record format affords more power in the data analysis step, as it provides multiple fields that correlation can be done on (unlike something like command histories). Both Lee, et al, and Neri participated

in the 1998 DARPA ID shootout [DARPA] that provided network data in raw packet form. Both found that converting the network data to connection logs aided performance with their data mining techniques [Lee:framework, Lee:dataflow, Lee:thesis, Neri]. Dickerson and Dickerson came to a similar conclusion. They found that compressing the data down to just the IP source, IP destination and the destination port, which they call the *sdp*, for each connection gave the best results [Dickerson].

Techniques for data mining

Given a wealth of connection logs, there are numerous ways to perform the actual mining for knowledge in them. The two primary techniques employed are statistical metrics and classification. There are numerous methods to do classification, we look at inductive rule generation, rule generation through inductive algorithms, fuzzy logic and neural networks. We will conclude with a selection of other proposed techniques.

Statistical techniques

Denning described how to use statistical measures to detect anomalies, as well as some of the problems and their solutions in such an approach. The five statistical measures that she described were the operational model, the mean and standard deviation model, the multivariate model, the Markov process model and the time series model [Denning]. She noted that the time series model was similar to the mean and standard deviation model in terms of applicability, and the time series model stood to provide more accurate results however it was more costly than the standard deviation model. It may be likely that the application of the time series model is more feasible in the off-line data mining environment, especially given the huge increase in computing power in the past 14 years. These statistical measure models may be further assisted by the availability of a data warehouse to do mining from as not all the metrics that are going to be used need to be known up front. If the system or analyst decides that the system should detect anomalies in the mean and standard deviation of duration of FTP sessions, the necessary mean and standard deviation can be constructed from the data warehouse and used promptly, rather than having to wait for the collection of new data.

Classification techniques

A classification based IDS attempts to classify all traffic as either normal or intrusive in some manner. The primary difficulty in this approach is how accurately the system can learn what the normal patterns are. This ultimately affects the accuracy of the system both in terms of whether real non-hostile activity is not flagged (false positive vs. false negative) and whether unusual activity will be flagged (true positive vs. true negative). Four different approaches have been tried to produce the patterns of normal activity given a proper training set: inductive rule generation, genetic algorithms, fuzzy logic and neural nets. We will look at each in turn.

Inductive rule generation, such as that done by RIPPER [Cohen] has been shown to be a fairly effective and straightforward way to produce a system that classifies traffic into normal and various forms of intrusive patterns [Lee:framework, Lee:dataflow, Lee:thesis]. The system is a set of rules than can be applied to the network traffic to classify it properly. One of the attractive features of this approach is that the rule set generated is easy to understand, hence a security analyst can verify it. A unique feature of this approach is the use of RIPPER to produce the set of features from the data to use in mining the data, in addition to the rules themselves (meta-learning) [Lee:framework, Lee:thesis]. The rules produced by RIPPER were too inefficient to apply directly in a real-time intrusion detection environment. Lee, et al, did some preliminary work in optimizing this rule set [Lee:dataflow], however there appear to remain some high-cost rules that would be better suited for off-line analysis.

The other classification approach is to produce a rule set via the application of a genetic algorithm using a system such as REGAL [Neri]. The approach is essentially identical to the inductive approach, although it does not use a separate meta-learning step. Additionally, Neri does not comment on the clarity or efficiency of the rules produced.

An interesting combination of the statistical metrics and classification approaches has been done using fuzzy logic. In [Dickerson] the authors classify portions of the data (where the portions are temporally related) based on various statistical metrics. They then create and apply fuzzy logic rules to these portions of data to classify it as normal or some type of intrusion. They found that the approach is particularly effective against

scans and probes [Dickerson]. It would appear that the primary disadvantage to this approach is the labor intensive rule generation process. Additionally, it would be instructive to see how well this approach works against intrusions other than scans and probes, particularly against other systems using a benchmark such as the DARPA IDS shootout data. Nevertheless, this work sparks ideas such as the potential power for a system based on fuzzy logic if the rules were generated automatically as in [Lee:framework, Lee:dataflow, Lee:thesis, Neri] or the use of fuzzy logic as a combinatorial tool (as opposed Bayesian statistics, belief networks or covariance matrices [Kumar]) to improve the accuracy of an entire system employing multiple approaches.

Ghosh, et al investigated the application of neural nets to classify network traffic. They began by using a basic signature matching system as a reference. They then attempted to use a pure feed-forward, backpropagation neural network. The best trained network performed similarly to the matching techniques, but its performance plateaued at a lower level. They then replaced the pure feed-forward neural net with an Elman network, which includes hidden context nodes. "Despite being the least extensively tuned of the three methods employed, the Elman nets produced the best results overall." This was attributed to the temporal state nature of network traffic [Ghosh].

Other techniques

Numerous other techniques have been suggested that could be used to mine security information from network connection logs. A technique that may improve the accuracy of detecting statistical anomalies is the generalized Markov chain. It has been noted that these are complex and time consuming to construct [Kumar], however their use may be possible in a high-power off-line environment. A technique that has been successfully applied to misuse detection systems is colored Petri nets [Kumar]. In IDIOT[fn1], the colored Petri nets were created by hand. It would be instructive to investigate if similar Petri nets could be constructed automatically and used for intrusion detection in an off-line environment. The Dempster-Shafer Method may be useful as a combinatorial tool in a system employing multiple approaches or fusing the data from multiple sources [Bass]. Finally, Lane identified numerous techniques from the signal processing and pattern recognition communities such as "spectral analysis, principle component analysis, linear regression, linear predictive coding, (gamma, epsilon)-similarity, neural networks, and nearest-neighbor matching" [Lane] that, while not well suited for data mining of command line histories as Lane was doing, may be better suited for network traffic analysis.

fn1: The way this author understands how IDIOT came about its name is that Dr Spafford, Kumar's advisor, told him that he needed to come up with a "snappy" name/acronym for his IDS. If he couldn't before a particular publication deadline, Spafford said the system would be named IDIOT for Intrusion Detection In Our Time. Obviously, Kumar did not come up with a better name and the name IDIOT stuck.

Results

One problem that is present in most of the above systems is the need for a properly labeled set of normal training data from which to construct standard statistical measures or train advanced systems. Statistical analysis requires such a set to ascertain what constitutes normal activity and classification based systems that can be trained require the set for their training. This problem hasn't been well addressed by the existing literature. The accepted practice seems to be to have a few weeks of training data that is throughly analyzed and labeled by a human analyst. Many of the research projects to date [Lee:framework, Lee:dataflow, Lee:thesis, Neri, Ghosh] have used the data provided as part of the 1998 DARPA intrusion detection system shootout [DARPA], which contains two sets of data. One set was for training which contains 14 labeled vulnerabilities and a second test set that contains 38 vulnerabilities, 24 of which have not been seen before. While such data sets are essential for comparing different IDS, they are highly labor intensive to produce and may not be totally accurate. Unfortunately, it is still necessary to produce such training sets for every environment where an IDS is used. One possible solution, as noted above, is for the IDS to train itself. This, however, leads to the problem of over specialization: strong behaviors (both good and bad) become stronger and weak behaviors eventually disappear.

The disadvantages aside, using the same dataset for analysis allows for a more quantitative comparison of the various IDSs discussed here. Of the three IDSs that used the 1998 DARPA dataset, two used the tcpdump portion of the data [Lee:thesis, Neri] and the third used only the BSM portion of the data [Ghosh]. We

provide a quantitative comparison of the results of the first two and the results of the third separately for qualitative comparison.

Anomaly detection based IDS, such as the ones examined here, tend to have a "sweet spot" where the number of true-positives outweighs the number of false-positives. The ratio of true-positives to all intrusions at this point is a good measure of the effectiveness of the system. Graphical representations of this can be found in [Lee:framework, Lee:dataflow, Lee:thesis, Neri]. Using these graphical representations we present a set of approximate[fn2] results from different types of intrusions in order to compare the RIPPER based (Inductive rule generation) and the REGAL based (Genetic algorithm rule generation) techniques (definitions of the intrusion categories can be found in [DARPA])[fn3]:

Intrusion Category	RIPPER (Induction)	REGAL (Genetic)
DOS	68%	80%
PROBING	97%	98%
U2R	78%	50%
R2L	19%	20%

fn2: Figures obtained from by interpretation of ROC graphs in [Lee:thesis, Neri].

fn3: We did not break out the differences between intrusion patterns that had been seen before and those that had not as Neri did not provide this information. For Lee's results in this area, see [Lee:thesis].

Ghosh, et al obtained impressive results in the U2R and R2L categories (only the aggregate of the two was reported) with their neural networks. Their best backpropagation neural network was able to achieve a 77.3% detection rate with a 2.2% false positive rate. The Elman net was able to achieve the 77.3% detection rate with no false positives and a 100% detection rate with less than 10% false positives.

Available products & services

Given all the research that is available, the authors find it surprising that there aren't more tools available either commercially or in the public domain. We examined the products that were available to do analysis of externally collected log information and identified fwlogwatch, logger, the WebTrends Firewall Suite, the GIAC network coordinated by SANS, ARIS, ISOA, DIDS, EMERALD, Kane Secure Enterprise and GrIDS. The first three are tools to do local analysis. The fourth is a service provided by SANS and their affiliates DShield, MyNetWatchman and NetSquared. The fifth is a similar service from Security Focus. The remaining products are all Intrusion Detection Systems. Many of these tools don't take much advantage of the potential analysis that can be done off-line, but each provides a useful capability that assists analysts in examining various types of log files.

Analysis products

RUS-CERT at the University of Stuttgart has developed the fwlogwatch tool [RUSCERT] which allows an analyst to search for different types of patterns in firewall logs in order to locate intrusions and intrusion attempts. Its queries are configurable to the extent that hosts and ports can be included or excluded as the analyst sees fit. fwlogwatch also has certain features, such as an integrated DNS resolver, to optimize performance. It detects time discrepancies in the log files which may indicate modification by an intruder in an attempt to cover their tracks. One of the features of fwlogwatch that is of particular interest to CSIRTs is the option for a user to generate a standardized report when an incident of interest is found.

Lance Spitzner has developed a Microsoft Access application called logger [Spitzner] that does basic analysis of Checkpoint-1 firewall logs. In particular, it allows an analyst to quickly pinpoint what IP addresses the firewall is rejecting most frequently.

The WebTrends Firewall Suite [WebTrends] is a commercial package to do analysis of firewall logs. It's primary feature is a user-friendly summary of content of the firewall log including a graph of bandwidth by

protocol and statistics on number of events, errors and bytes transfered.

Services

One of the more promising projects taking place is the Consensus Intrusion Database (CID) Project [CID], which is being done by the SANS Global Incident Analysis Center (GIAC) [GIAC] in conjunction with it's partners DShield [DShield], MyNetWatchman [MyNetWatchman], and NetSquared [NetSquared]. These partner sites collect, correlate and analyze firewall logs. Based on this analysis the partners identify what systems are performing undesirable activity against other systems on the Internet. A list of these systems is then forwarded to CID. Administrators can then use the CID list to identify if any of those undesirable systems attempted to connect to their systems, and even block those systems if the administrator's are so inclined.

The most open of the partner sites is DShield [DShield]. DShield accepts various types of firewall log files from any source and allows anyone to view activity summaries such as the IP addresses that are doing the most negative activity or the ports that are being scanned for most often. DShield also provides a general query that allows you to query their database for a given address (either source or destination), source and destination ports, IP protocol and date ranges. Given the wealth of data they have (over a million log entries after collecting data for about 3 months), the information obtained from the queries may be very useful to an analyst.

Another similar site that started recently is the Attack Registry & Intelligence Service (ARIS) from Security Focus [ARIS]. ARIS does much less cross-site correlation and analysis than CID does. ARIS simply serves as a registry for incident information and allows system administrators a convenient way to do analysis and obtain summary reports like what addresses are hitting your systems the hardest.

Intrusion detection systems

There are numerous intrusion detection systems that implement various aspects of the system outlined above to do data fusion and mining of data to find security information. These systems range from research oriented systems to commercial products. We have omitted those already mentioned in the section on research and will focus on the IDSs that are closest to the system described above or have been particularly pivotal or unique in the development of IDSs. These IDSs are ISOA, DIDS, EMERALD, Kane Secure Enterprise and GrIDS.

The first couple of IDS of record [CERIAS] that performed data fusion and cross sensor correlation were the Information Security Officer's Assistant (ISOA) [ISOA1, ISOA2] and the Distributed Intrusion Detection System (DIDS) [DIDS1, DIDS2]. ISOA conglomerated the audit information for numerous hosts whereas DIDS conglomerated the audit information from numerous host and network based IDSs. Both used a rules based expert system to perform the centralized analysis, although ISOA was more focused on anomaly detection and DIDS on misuse detection. Additional features of note were that ISOA provided a suite of statistical analysis tools that could be employed either by the expert system or a human analyst, and DIDS expert system featured a limited learning capability.

EMERALD extended some of the seminal IDS work at SRI [Denning, NIDES] with a hierarchical analysis system: the various levels (host, network, enterprise, etc) would each perform some level of analysis and pass any interesting results up the chain for correlation [EMERALD1, EMERALD2]. There was a feedback system in place so that the higher levels could request more information for a given activity. Of particular interest to us was the analysis done at the top level which monitored the system for "network-wide threats such as Internet worm-like attacks, attacks repeated against common network services across domains, or coordinated attacks from multiple domains against a single domain," [EMERALD1]. The EMERALD architects employed numerous approaches such as statistical analysis, an expert system and modular analysis engines as they believed, "no one paradigm can cover all types of threats. Therefore we endorse a pluralistic approach," [EMERALD2].

Kane Secure Enterprise is a commercial product that performs data fusion for numerous types of devices (host audit information, firewall logs, and host and network based IDSs) and performs both misuse and anomaly detection on the results [Kane]. It is unique in the realm of IDS by virtue of operating both in a

real-time and in an off-line capacity (as appropriate for the given analysis). All data is stored in a relational database as it comes in. While some simple signatures are matched at this time and generate an alert, much of the analysis is done off-line. This analysis is done with both statistical analysis and a CLIPS based expert system. There is apparently some type of correlation analysis done on the data, unfortunately Kane doesn't provide any details on the techniques employed. Finally, Kane provides "a complete set of tools and reports for database analysis and investigation" by a human analyst.

The final IDS of interest is GrIDS - A Graph based Intrusion Detection System for large networks [GrIDS]. GrIDS creates graphs of network activity which reveal the causal structure of the network traffic which allows coordinated attacks to be easily detected. This system is of interest primary as it successfully implements a technique that should be useful to a system performing data mining for security information.

Other work

It is interesting to note that there have been no patent claims for any system alleging to apply data mining techniques to find any type of computer security information. There are numerous organizations that sell some form of a networking monitoring service. None of these organizations publicly disclose to what extent they perform data mining on the log information they collect. This is most likely because they consider that information to be a trade secret that gives them a competitive advantage.

Conclusion

We have outlined what data mining is with respect to computer security, along with a vision of what an ideal off-line system for data mining for security information would be capable of. We have presented the existing research in this area with results where possible. Additionally, we looked at promising techniques for future investigation. We then examined products that were available that implement some select features of the detailed ideal system. These products ranged from log analyzers to web based analysis centers to various IDSs. Finally, we provided a very brief overview of other work in this field.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

References

1. Bass: Bass, "Intrusion Detection Systems and Multisensor Data Fusion", Communications of the ACM, Vol. 43, No. 4, pp 99-105, April 2000
2. Lee:framework: Lee, Stolfo & Mok, "A Data Mining Framework for Building Intrusion Detection Models", Proceedings of the 1999 IEEE Symposium on Security and Privacy, pp 120-132, 1999, IEEE
3. Waltz: Waltz & Llinas, *Multisensor Data Fusion*, 1990, Artech House
4. T&C: Thuraisingham & Ceruti, "Understanding Data Mining and Applying it to Command, Control, Communications and Intelligence Environments", The 24th Annual International Computer Software and Applications Conference (COMPSAC), pp 171-175, 2000, IEEE
5. Lee:Thesis: Lee, *A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems* , Doctoral Dissertation, Columbia University, 1999
6. Ghosh: Ghosh, Schwartzbard & Schatz, "Learning Program Behavior Profiles for Intrusion Detection", Proceedings of the Workshop on Intrusion Detection and Network Monitoring, April 1999, USENIX
7. Kumar: Kumar, *Classification and Detection of Computer Intrusions*, Doctoral Dissertation, Purdue University, 1995
8. Denning, "An Intrusion-Detection Model", IEEE Transactions on Software Engineering, Vol. 13, No. 2, pp 222-232, February 1987
9. craw: http://slashdot.org/comments.pl?sid=01/03/23/1850206&cid=214
10. Lane: Lane, *Machine Learning Techniques for the Computer Security Domain of Anomaly Detection*, Doctoral Dissertation, Purdue University, August 2000
11. UNM: http://www.cs.unm.edu/~immsec/research.htm

12. Dickerson: Dickerson & Dickerson, "Fuzzy Network Profiling for Intrusion Detection", 19th International Conference of the North American Fuzzy Information Processing Society, 2000, pp 301-306, IEEE
13. DARPA: Lippmann, Fried, Graf, Haines, Kendall, McClung, Weber, Webster, Wyschogrod, Cunningham, Zissman, "Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation", Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX), pp 12-26, 2000, IEEE
14. Lee:dataflow: Lee, Stolfo, Mok, "Mining in a data-flow environment: experience in network intrusion detection", Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 114-124, 1999, ACM
15. Neri, "Comparing local search with respect to genetic evolution to detect intrusions in computer networks", Proceedings of the 2000 Congress on Evolutionary Computation, Vol. 1, pp 238-243, IEEE
16. Cohen: Cohen, "Fast effective rule induction", Machine Learning: the 12th International Conference, 1995, Morgan Kaufmann
17. RUSCERT: http://cert.uni-stuttgart.de/projects/fwlogwatch/
18. Spitzner: http://www.enteract.com/~lspitz/logger.html
19. WebTrends: http://www.webtrends.com/products/firewall/default.htm
20. CID: http://cid.fearnow.net/
21. GIAC: http://www.sans.org/giac.htm
22. DShield: http://www.dshield.org/
23. MyNetWatchman: http://www.mynetwatchman.com/
24. NetSquared:http://www.netsquared.com/
25. ARIS: http://aris.securityfocus.com/
26. CERIAS: http://www.cerias.purdue.edu/coast/intrusion-detection/ids.html
27. ISOA1: Winkler, Page, "Intrusion and anomaly detection in trusted systems", Fifth Annual Computer Security Applications Conference, 1989, 1990, pp 39-45, IEEE
28. ISOA2: jtruitt@dw3f.ess.harris.com, "PRC's Information Security Officer's Assistant (ISOA)", posting to the IDS mailing list, August 1994, http://www.geek-girl.com/ids/1994/0028.html
29. DIDS1: Snapp, Brentano, Dias, Goan, Heberlein, Ho, Mukherjee, "A System for Distributed Intrusion Detection". COMPCON Spring '91, Digest of Papers, 1991, pp 170-176
30. DIDS2: Snapp, Brentano, Dias, Heberlein, Ho, Levitt, Mukherjee, "DIDS -- Motivation, Architecture, and an Early Prototype", Proceedings of the 14th National Computer Security Conference, October 1991, pp 167-176
31. NIDES: http://www.sdl.sri.com/projects/nides/
32. EMERALD1: Porras, Neumann, "EMERALD: Conceptual Overview Statement", December 1996, http://www.sdl.sri.com/papers/emerald-position1/
33. EMERALD2: Neumann, Porras, "Experience with EMERALD to Date", First USENIX Workshop on Intrusion Detection and Network Monitoring, 1999, pp 73-80
34. Kane: http://www.intrusion.com/product/product.asp?lngProdNmId=3
35. GrIDS: Staniford-Chen, Cheung, Crawford, Dilger, Frank, Hoagland, Levitt, Wee, Yip, Zerkle, "GrIDS - A Graph Based Intrusion Detection System for Large Networks", NISSC 1996 http://olympus.cs.ucdavis.edu/arpa/grids/welcome.html